ARL-FLIGHT-MECH-TM-410 AR-005-598

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORY

MELBOURNE, VICTORIA

Flight Mechanics Technical Memorandum 410

A FORTRAN PROGRAM FOR THE CALCULATION OF THE CALIBRATION COEFFICIENTS OF A SIX-COMPONENT STRAIN GAUGE BALANCE

by

S.S.W. Lam

Approved for Public Release

FEBRUARY 1989

AR-005-598

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
AERONAUTICAL RESEARCH LABORATORY

Flight Mechanics Technical Memorandum 410

A FORTRAN PROGRAM FOR THE CALCULATION OF THE CALIBRATION COEFFICIENTS OF A SIX-COMPONENT STRAIN GAUGE BALANCE

by

S.S.W. Lam

SUMMARY

A computer program written in FORTRAN has been developed to calculate the calibration coefficients of a six-component strain gauge balance. The method of Least Squares is used in the calculation procedures which do not require the components of the balance to be loaded independently. The procedures are, in fact, independent of the way the balance is loaded as long as the supplied load/output data provide sufficient information for the determination of the coefficients. The program incorporates several statistical estimations for the evaluation of the performance of the calibration process. It can therefore be used as an experimental tool for the investigation of different loading schemes which may be used to calibrate a strain gauge balance.

POSTAL ADDRESS: Director, Aeronautical Research Laboratory,
P.O. Box 4331, Melbourne, Victoria, 3001, Australia

Contents

Nomenclature

$[A]$	a matrix derived from the Least Squares criteria, defined by equation (7)
a_{ij}	coefficient of partial correlation
$C_{i,j}$	coefficient of calibration
$[C]$, $[C1]$, $[C2]$	calibration coefficient matrices
$[E]$	a matrix derived from the Least Squares criteria, defined by equation (5)
H_i	a component load of strain gauge balance
$[H]$	vector of component loads
$[H^*]$	vector of "non-linear" loading terms
N	total number of sets of data
N_f	degree of freedom in statistical estimates
p	index of summation
R_i	a component of strain gauge balance output
r_i	coefficient of multiple correlation
$S_{H,i}$	standard deviation of H_i
$S_{RH,i}$	standard error of estimates
X_i	a variable in regression equation

1 Introduction

In wind tunnel operations, calibrating a multi-component strain gauge balance is a process of determining the coefficients of an equation which describes the relationship between the applied loads and the strain gauge outputs from each component. This involves choosing an appropriate calibration equation which best describes the behaviour of the particular balance. The balance is then loaded in a specific manner and the output from each strain gauge component is noted. A sufficient number of loadings has to be applied to the system, depending on the complexity of the chosen calibration equation, in order that the coefficients of the calibration equation may be totally determined.

Ideally, a perfect strain gauge balance would have each strain gauge output corresponding to one and only one load component. However, in a real situation, it is impossible to completely eliminate interactions from other components. Very often, significant non-linear interactions (functions of a combination of load components) are present as well as the usual linear ones (functions of a single load component). For a well designed 6-component balance, it is usual to express the strain gauge output of the ith balance component (R_i, $i = 1, ..., 6$) as a second order polynomial function of all the components of the applied load (H_j, $j = 1, ..., 6$). Thus the form of the calibration equation of a 6-component strain gauge balance is

$$\begin{aligned} R_i \;=\; & C_{i.1}H_1 + C_{i.2}H_2 + \cdots + C_{i.6}H_6 \\ & + C_{i.11}H_1^2 + C_{i.22}H_2^2 + \cdots + C_{i.66}H_6^2 \\ & + C_{i.12}H_1H_2 + C_{i.13}H_1H_3 + \cdots + C_{i.56}H_5H_6, \qquad i = 1, ..., 6 \end{aligned} \tag{1}$$

where the C's are the calibration coefficients determined during balance calibration and H_i are the component loads. The calibration coefficients of each component i may be classified as follows:

1. "linear", e.g. $C_{i.j}$ for $j = 1, ..., 6$;
2. "load squared", e.g. $C_{i.jj}$ for $j = 1, ..., 6$;
3. "load cross product", e.g. $C_{i.jk}$ for $j = 1, \ldots, 5$, and $k = (j + 1), ..., 6$.

However, due to the difference in characteristics of a strain gauge in tension and in compression, an odd function representation is more appropriate to describe its behaviour on both sides of the origin. Hence the load/output relationship of a strain gauge balance may better be accounted for by the inclusion of the "load cubed" terms in the calibration equation. Thus the right hand side of equation (1) is modified with the addition of the terms

$$C_{i.111}H_1^3 + C_{i.222}H_2^3 + \cdots + C_{i.666}H_6^3.$$

The set of six calibration equations for the six components can be represented in matrix form, with the calibration matrix partitioned into "linear" and "non-linear" sub-matrices:

$$[R] = [C1][H] + [C2][H^*] \tag{2}$$

where $[R]$ is the vector of strain gauge balance component outputs,
$[C1]$ is the matrix of linear calibration coefficients, $C_{i,j}$,
$[C2]$ is the matrix of non-linear calibration coefficients, $C_{i,jj}$, $C_{i,jk}$ and $C_{i,jjj}$,
$[H]$ is the vector of component loads, and
$[H^*]$ is the vector of squares, cross products and cubes of component loads.

The conventional method of calibrating a strain gauge balance of the type given by equation (2) is to apply pure component loads independently. When a single component is loaded, the outputs of all components of the strain gauge balance respond to one and only one load component. The "linear", "load squared" and the "load cubed" coefficients may hence be determined. The "load cross product" coefficients are found by applying two pure component loads simultaneously to the balance keeping one load constant at a finite value while varying the other. This method, although straightforward to use, relies on pure component loads which rarely represent the real, practical loading situation which the model encounters. It contravenes the basic principle that the loadings of the balance during the calibration should be representative of that experienced by the model during wind tunnel tests.

Theoretically, if the calibration equation describes exactly the relationship between the applied load and the strain gauge output, the way in which the balance is loaded during calibration should not affect the outcome of the results. Whether the balance components are loaded independently or simultaneously, the values of the calibration coefficients should be the same. However, in reality, the calibration equation given by equation (2) is only an approximation and does not describe *exactly* the load/output relationship of a practical strain gauge balance. The responses of the strain gauge balance components can be quite different under different loading distributions which causes different deflection characteristics in the balance. The calibration coefficients thus calculated are dependent of how the balance is loaded during the calibration.

Ramaswamy et al. [1] proposed a Least Squares (LS) method for the calibration of wind tunnel balances. This method does not require the balance components to be loaded independently. In fact, any combination of component loads may be applied simultaneously. The distributions of loads on the balance during the calibration can be chosen to match as closely as possible those likely to be experienced by the model during wind tunnel tests. In addition, there is no restriction of the sequence or pattern in which the loads have to be applied in order to obtain a particular calibration coefficient. Virtually any *random* set of component loads may be applied to the balance in any sequence provided that all the data so obtained generate enough independent information for the calculations of the coefficients. Another advantage of the LS method is that the "linear" and "non-linear" coefficients can be determined at the same time. This is because by loading all the components simultaneously, the "linear" and "non-linear" interactions among the components are all accounted for in the data. Less effort will hence be required to calibrate the balance if one can design an optimum loading scheme which would simulate the real loading situation as well as providing the most

information for the determination of the coefficients with the least number of sets of loads.

A major shortcoming of the LS method is that it assumes the independent variables, i.e. the H's in equation (1), are free of noises. In reality, due to certain factors such as friction or misalignment of the point of application of H_i, even though the same load is applied to the balance twice, the actual load experienced by the balance may not be exactly the same in both cases. Nevertheless, the errors arising from the difference between the load applied and the actual load experienced by the balance are usually insignificant when compared with the noises in the outputs R_i. Furthermore, for a well designed strain gauge balance, the noises in R_i are often satisfactorily small such that the LS method can normally be applied successfully.

This report describes a computer program, `SGBCalib`, which implements the LS method of determining the coefficients of the calibration equation given by (2). The only inputs to the program are the sets of applied component loads and the corresponding strain gauge outputs. If the given data contain sufficient information, the calibration coefficients will be evaluated. The program also generates useful statistical estimates of how closely the resulting calibration equation fits the given data. The reliability of the calibration equation is expressed in terms of the standard error of estimate of the strain gauge outputs for all the six components. The 95% confidence levels of the "linear" coefficients are also estimated as indications of the accuracy of the calibration.

Apart from the function of calculating the calibration coefficients from a given set of data, `SGBCalib` can be used to assist in the investigation of how the balance components should be loaded so as to yield the most information with the least number of sets of applied loads and at the same time best simulate the patterns of loading expected in the wind tunnel testings. It can also be used for studies of whether the errors of a particular strain gauge output could be reduced by further loadings designed to optimally minimize the uncertainties in the calibration coefficient.

2 Formulations of the Least Squares Method

Given that for a set of component loads, H_i where $i = 1, ..., 6$, the corresponding strain gauge balance outputs are R_i. If there are N sets of H_i with N sets of corresponding R_i, denoted by $H_{i,p}$ and $R_{i,p}$ where $p = 1, ..., N$, then the calibration coefficients may be found such that the sum of the squares of the difference between the measured strain gauge output and that obtained from the calibration equation is a minimum. That is

$$e_i = \sum_{p=1}^{N} \left[C_{i,1} H_{1,p} + C_{i,2} H_{2,p} + \cdots + C_{i,666} H_{6,p}^3 - R_{i,p} \right]^2 , \quad i = 1, ..., 6 ,$$

should be a minimum. This involves partial differentiating each of the e_i with respect to each of the coefficients and equating the resulting expressions to zero. The result is a set of 33

non-linear simultaneous equations for each component i given as follows[1]:

$$\begin{aligned}
\sum \left[C_{i.1}H_1 + C_{i.2}H_2 + \cdots + C_{i.666}H_6^3 - R_i\right] H_1 &= 0 \\
\sum \left[C_{i.1}H_1 + C_{i.2}H_2 + \cdots + C_{i.666}H_6^3 - R_i\right] H_2 &= 0 \\
&\vdots \\
\sum \left[C_{i.1}H_1 + C_{i.2}H_2 + \cdots + C_{i.666}H_6^3 - R_i\right] H_6^3 &= 0.
\end{aligned} \tag{3}$$

It is more convenient to put equations (3) in matrix form:

$$[E][C_i] = [A_i]\,, \qquad i = 1, ..., 6\,, \tag{4}$$

where

$$[E] = \begin{bmatrix} \sum H_1H_1 & \sum H_1H_2 & \cdots & \sum H_1H_6^3 \\ \sum H_2H_1 & \sum H_2H_2 & \cdots & \sum H_2H_6^3 \\ \vdots & \vdots & & \vdots \\ \sum H_6^3H_1 & \sum H_6^3H_2 & \cdots & \sum H_6^3H_6^3 \end{bmatrix} \tag{5}$$

is a 33 × 33 square matrix and is common for all the six components. The two column matrices, $[C_i]$ and $[A_i]$, for the ith component are given by

$$[C_i] = \begin{bmatrix} C_{i.1} \\ C_{i.2} \\ \vdots \\ C_{i.666} \end{bmatrix} \qquad\qquad [A_i] = \begin{bmatrix} \sum H_1R_i \\ \sum H_2R_i \\ \vdots \\ \sum H_6^3R_i \end{bmatrix}.$$

The $[C_i]$'s and $[A_i]$'s for all the six components may be combined to form two 33 × 6 matrices, $[C]$ and $[A]$, such that

$$[C] = \begin{bmatrix} C_{1.1} & C_{2.1} & \cdots & C_{6.1} \\ C_{1.2} & C_{2.2} & \cdots & C_{6.2} \\ \vdots & \vdots & & \vdots \\ C_{1.666} & C_{2.666} & \cdots & C_{6.666} \end{bmatrix}, \tag{6}$$

$$[A] = \begin{bmatrix} \sum H_1R_1 & \sum H_1R_2 & \cdots & \sum H_1R_6 \\ \sum H_2R_1 & \sum H_2R_2 & \cdots & \sum H_2R_6 \\ \vdots & \vdots & & \vdots \\ \sum H_6^3R_1 & \sum H_6^3R_2 & \cdots & \sum H_6^3R_6 \end{bmatrix}. \tag{7}$$

[1] For the sake of clarity, the subscript p for R_i and H_i is omitted in the remainder of the report. Whenever the summation symbol is used, it is understood that the sum is to be taken over $p = 1, ..., N$.

The Least Squares criteria can therefore be represented by the matrix equation

$$[E][C] = [A] . \tag{8}$$

Provided that $[E]$ is non-singular, the coefficient matrix $[C]$ may be found as

$$[C] = [E]^{-1}[A] . \tag{9}$$

3 Statistical Estimations

In this section we adopt a few statistics to measure the accuracy and reliability of the calibration of the balance using the LS method. We seek to determine how well the resulting calibrating equation describes the relationship between the strain gauge outputs and the applied component loads. The statistics described below provide us with information on and measurements of the goodness of fit of the calibration equation, and the uncertainties of the coefficients determined by the LS method.

3.1 Standard Error of Estimates

The standard error of estimates of the ith component is given by

$$S_{RH,i} = \sqrt{\frac{\sum(R_i - \hat{R}_i)^2}{N_f}} \tag{10}$$

where R_i is the measured ith component of the strain gauge balance output,
$\hat{R}_i$ is the corresponding estimated value using the calibration equation,
N_f is the degree of freedom, which is $(N - 33)$, and
N is the total number of sets of loading cases.

It is a measure of the scatter of the balance outputs about the estimated values. It has properties analogous to those of the standard deviation, i.e., if N is large enough, there would be approximately 68%, 95% and 99.7% of the measured R_i lies within the bounds of $S_{RH,i}$, $2S_{RH,i}$ and $3S_{RH,i}$ respectively from the estimated $\hat{R}_i$ value.

3.2 Coefficient of Multiple Correlation

The coefficient of multiple correlation of the ith component, r_i, is defined as

$$r_i = \sqrt{\frac{\sum(\hat{R}_i - \bar{R}_i)^2}{\sum(\hat{R}_i - \bar{R}_i)^2 + \sum(R_i - \hat{R}_i)^2}} \tag{11}$$

The term $\sum(\hat{R}_i - \bar{R}_i)^2$ is known as the *explained variation*, so called because it accounts for the variations (of R_i from the mean) which are explained by the regression. $\sum(R_i - \hat{R}_i)^2$ is the *unexplained variation*. It represents the variations which remain unexplained after the regression. The coefficient of multiple correlation shows what fraction of the overall variations is accounted for by applying the calibration equation. It is therefore a measure of the correlation between a balance component output and all of the component loads. It shows how well the calibration equation describes the relationship between the outputs of the strain gauge balance and the component loads. It is also a measure of the closeness of fit of the calibration equation to the actual measured strain gauge balance outputs. The value of r_i lies between 0 and 1. The closer it is to 1 the better is the linear relationship. When $r_i = 1$ the correlation is called *perfect*.

3.3 Coefficient of Partial Correlation

The coefficient of partial correlation measures the linear relationship between a component of the strain gauge balance output and a single component load while the other component loads are kept constant.

The linear correlation coefficients a_{12} between two variables X_1 and X_2 is given by

$$a_{12} = \frac{N\sum X_1X_2 - (\sum X_1)(\sum X_2)}{\sqrt{[N\sum X_1^2 - (\sum X_1)^2][N\sum X_2^2 - (\sum X_2)^2]}} \tag{12}$$

where N is the number of data points. a_{12} is called the zero order correlation coefficient. We can denote the coefficient of partial correlation between X_1 and X_2 keeping X_3 constant by $a_{12.3}$ (the 1st order correlation coefficient) such that

$$a_{12.3} = \frac{a_{12} - a_{13}a_{23}}{\sqrt{(1 - a_{13}^2)(1 - a_{23}^2)}}. \tag{13}$$

Similarly the coefficient of partial correlation between X_1 and X_2 keeping X_3 and X_4 constant may be given by (see for e.g. Spiegel [2])

$$a_{12.34} = \frac{a_{12.4} - a_{13.4}a_{23.4}}{\sqrt{(1 - a_{13.4}^2)(1 - a_{23.4}^2)}} \cdot \frac{a_{12.3} - a_{14.3}a_{24.3}}{\sqrt{(1 - a_{14.3}^2)(1 - a_{24.3}^2)}}. \tag{14}$$

The coefficient of partial correlation between the strain gauge balance output R_i and the component load H_1, say, keeping the other component loads constant may therefore be denoted by $a_{R_i1\,23456}$, where the numerical subscripts denote the component loads, and can be deduced from equations (12), (13) and (14).

An ideal strain gauge balance would have perfect linear correlation ($a = 1$) between an output and its corresponding component load and zero correlation with the rest of the components. These coefficients therefore indicate how far the strain gauge balance has fallen short of this ideal behaviour.

3.4 95% Confidence Interval

To establish a 95% confidence interval for a calibration coefficient $C_{i,j}$ we can construct the t-distribution statistics:

$$t = \frac{C_{i,j} - \bar{C}_{i,j}}{S_{RH,i}/S_{H,j}}\sqrt{N_f} \qquad i = 1,\ldots,6,\ \ j = 1,\ldots,6$$

where $S_{RH,i}$ is the standard error of estimate of the ith component given in section 3.1,
$S_{H,j}$ is the standard deviation of H_j,
$\bar{C}_{i,j}$ is the expected mean of $C_{i,j}$.

The 95% confidence interval for the coefficient $C_{i,j}$ is then given by the expression

$$\frac{t_{0.975}}{N_f}\left(\frac{S_{RH,i}}{S_{H,j}}\right).$$

The percentile values $t_{0.975}$ for various degree of freedom (N_f) may be found in most reference books in statistics (see e.g. Yamane [3]).

4 The Computer Program SGBCalib

The program SGBCalib was written in FORTRAN and was implemented in DEC[1] MicroVAX[1] II operating under the VMS[1] (Version 4.5) system. The program, however, was not intended to be VMS-specific, and the language used conforms to the American National Standard FORTRAN-77 (ANSI X3.9-1978).

4.1 Program Variables

Drag measurements using wind tunnel force balance usually require high order of accuracy. In order to maintain data integrity, all the program variables concerned are assigned as "double precision". Most of the variables were passed between the various subroutines via common blocks:

/matrixA/, /matrixC/, /matrixE/, /inv/
: These common blocks contain the elements of the matrices $[A]$, $[C]$, $[E]$ and $[E]^{-1}$ respectively.

/loads/, /readings/

[1]DEC, MicroVAX and VMS are registered trademarks of Digital Equipment Corporation

They store all the given H_i and R_i data in the 2-D arrays HH and RR respectively. The program at present allows a maximum of 200 sets of H_i and R_i values to be processed. This may be changed if necessary by altering the dimension of the second subscripts of the statement

```
double precision   HH(6,200), RR(6,200).
```

The dimension of the second subscript of the variable res(6,200) in the subroutines Statistics, SumStats and Chart should also be changed to be the same as those of HH and RR.

/Hvector/ This is an array of the 33 H terms in the calibration equation. That is, it contains the values of

$$H_1, H_2, \ldots, H_6, H_1^2, H_2^2, \ldots, H_6^2, H_1H_2, H_1H_3, \ldots, H_5H_6,$$
$$H_1^3, H_2^3, \ldots, H_6^3.$$

/residual/ This common block appears in the statistical routines only. It consists of the residual values of R_i, i.e. $(R_i - \hat{R}_i)$. The block also contains the maximum value of the residual, resmax, which is used for scaling of the chart of the residuals.

/misc/ This is a miscellaneous block containing the pth set of values of R_i, $\hat{R}_i$ and H_i. The mean values of R_i are also included in this block.

/statsvar/ This block contains the variables used for the calculations of the various statistical estimates for the calibration. The major variables are:

sRH(i,j)	=	$\sum \hat{R}_i H_j, \quad i,j = 1,...,6$
sR(i)	=	$\sum \hat{R}_i$
sR2(i)	=	$\sum \hat{R}_i^2$
sH(i)	=	$\sum H_i$
sH2(i)	=	$\sum H_i^2$
sHH(i,j)	=	$\sum H_i H_j, \quad i,j = 1,...,6$
dRRe2(i)	=	$\sum (R_i - \hat{R}_i)^2$
dRRm2(i)	=	$\sum (R_i - \bar{R}_i)^2$
dReRm2(i)	=	$\sum (\hat{R}_i - \bar{R}_i)^2$

4.2 Program Structure

The program is made up of different subroutines designated for various specific tasks. The listing of these subroutines is given in Section 5. The tasks undertaken by these subroutines may be divided broadly into three areas:

1. Program inputs and outputs.

2. Calculations of the matrices $[A]$, $[E]$, $[E]^{-1}$ and $[C]$.

3. Estimation of the statistics of the calibration.

Apart from the various subroutines written specifically for the LS method of calculating the calibration coefficients, there are two main subroutines which are used in matrix operations. They are the matrix inversion (`MatInv`) and the matrix multiplication (`MatMult`) routines. For the sake of completion, these subroutines are also listed in Section 5.

4.2.1 Program Inputs and Outputs

The only program inputs are the sets of H_i and R_i values which are read from the data file `SGBIN.DAT`. The format of the file is as follows:

1. The first line is empty and is skipped by the program. It may be used for data identification purpose.

2. The second line contains a row of 6 components of loads, H_i, each of which is separated from the next value by one or more spaces.

3. The third line is a row of 6 components of strain gauge outputs, R_i, corresponding to each of the H_i value in the above row.

4. This pattern: an empty line, a row of 6 H_i values followed by a row of 6 corresponding R_i values, repeats throughout the data file for as many sets of loading as are available.

When the program is executed, all the relevant information and results are written o an output file `SGBOUT.DAT` which may subsequently be printed on a line printer. The outpu: of the program consist of the following:

1. All the values of H_i and R_i provided in the input file.

2. The evaluated calibration coefficient matrix $[C]$ in the form:

$$\begin{array}{cccccc}
C_{1,1} & C_{2,1} & C_{3,1} & C_{4,1} & C_{5,1} & C_{6,1} \\
C_{1,2} & C_{2,2} & \cdots & \cdots & \cdots & C_{6,2} \\
\vdots & \vdots & & & & \vdots \\
C_{1,11} & C_{2,11} & \cdots & \cdots & \cdots & C_{6,11} \\
\vdots & & & & & \vdots \\
C_{1,56} & C_{2,56} & \cdots & \cdots & \cdots & C_{6,56} \\
\vdots & & & & & \vdots \\
C_{1,666} & C_{2,666} & \cdots & \cdots & \cdots & C_{6,666}
\end{array}$$

3. Statistical estimations which include:

   - the standard errors of estimates for each of the component,
   - the coefficients of multiple correlation,
   - the coefficients of partial correlation between R_i and each of the component loads,
   - the 95% confidence level of the linear coefficients, i.e.

$$\begin{matrix} C_{1.1} & C_{2.1} & C_{3.1} & C_{4.1} & C_{5.1} & C_{6.1} \\ C_{1.2} & C_{2.2} & \cdots & \cdots & \cdots & C_{6.2} \\ \vdots & \vdots & & & & \vdots \\ C_{1.6} & C_{2.6} & C_{3.6} & C_{4.6} & C_{5.6} & C_{6.6} \end{matrix}$$

4. A table of comparisons of the measured and estimated values of R_i and the residuals.

5. A line printer plot of the residuals of R_i over $\hat{R}_i$ for each of the 6 components. The maximum deviation is ± 10 characters from the zero marker (the character "!") and is scaled to be equivalent to the maximum residual value of all 6 components.

4.2.2 Evaluations of the Matrices

As each set of the H_i and R_i values are read from the input file, the "load squares" (H_i^2), "load cubes" (H_i^3) and "load cross products" (H_iH_j, $i = 1,\ldots,5$ and $j = i+1,\ldots,6$) are evaluated by means of the subroutine `Hfactors`. The elements forming the matrices $[E]$ and $[A]$ are then calculated and summed in the subroutine `SumElements`.

After all the H_i and R_i values have been read, $[E]^{-1}$ is found using the matrix inversion routine `MatInv`. An error flag (designated by the variable `error`) is used to indicate if an inverse has successfully been calculated. When `error` = 0, the inverse is found and no error occurs. When `error` = 2, $[E]$ does not have an inverse and the program terminates with the message

```
%%% INVERSE OF [E] DOES NOT EXIST %%%
```

appearing on the terminal screen. When $[E]^{-1}$ is found the coefficient matrix $[C]$ is evaluated according to equation (9) by the matrix multiplication routine `MatMult` and the elements of $[C]$ are written to the output file `SGBOUT.DAT`.

4.2.3 Statistical Estimations

The evaluations of the various statistics mentioned in Section 3 are initiated by the subroutine `Statistics`. The estimated value of R_i, denoted as `Re(i)` in the program, is calculated

according to equation (2). Before equation (2) can be applied, the coefficient matrix $[C]$ is transposed and partitioned into a linear matrix $[C1]$ and a non-linear matrix $[C2]$. This is accomplished by the statements

```
do i = 1, 6
  do j = 1, 6
    c1(i,j) = c(j,i)
  end do
  do j = 1, 27
    c2(i,j) = c(j+6,i)
  end do
end do
```

in the `Statistics` routine. The subroutine `SGBReading` then evaluates `Re(i)` from the data of H_i according to equation (2) and the various summations required by the statistics are calculated in the subroutine `SumStats`.

The various statistics described in Section 3 are designated as

`S(i)` for the standard error of estimates $S_{RH,i}$,
`cmc(i)` for the coefficients of multiple correlation r_i, and
`c1(i,j)` for the 95% confidence levels of the linear coefficients $C_{i,j}$.

Because of the complexity of the calculations of the coefficients of partial correlation, they are described in more detail in Section 4.2.4. The calculation of the confidence levels requires the 0.975 percentile values at various degree of freedom for the t-statistics. These values are provided as a look-up table in the subroutine `Tstats`.

A graphical representation of the residuals of R_i over $\hat{R}_i$ is constructed by the subroutine `Chart`. The results for all six components are displayed together across the width of the page. The zero axes are represented by the character "!". The residuals for components 1 and 4 are represented by the character "*", those for components 2 and 5 by the character "o" and those for components 3 and 6 by the character "#". The scale of the deviation is based on the maximum value of the residual (designated by the variable `resmax` in the program) of all six components. The value of `resmax` occupies a space of 10 character on the chart and the other values of the residual are scaled accordingly.

4.2.4 Evaluation of Coefficient of Partial Correlations

For the sake of clarity, let us consider a single component of the strain gauge balance output R_3, say, and seek to determine the correlation coefficients between R_3 and each of the component loads H_i keeping the other component loads constant. It is convenient, then, to consider

an array X_i, $i = 1, \ldots, 7$ such that $X_1 = R_3$, $X_2 = H_1$, $X_3 = H_2$, ..., and $X_7 = H_6$. The coefficients of partial correlation between R_3 and each of the H's may therefore be denoted as

$$a_{12.34567},\ a_{13.24567},\ a_{14.23567},\ a_{15.23467},\ a_{16.23457},\ \text{and}\ a_{17.23456}.$$

To determine, say, $a_{12.34567}$, we may use the relationship

$$a_{12.34567} = \frac{a_{12.4567} - a_{13.4567}a_{23.4567}}{\sqrt{(1 - a^2_{13.4567})(1 - a^2_{23.4567})}} \tag{15}$$

in which the variable X_3 is used as the "mediator". Theoretically, we can use any of the "kept-constant" variables X_3, X_4, X_5, X_6 and X_7 to be the mediator to calculate $a_{12.34567}$. For convenience in programming, we opt for the lowest subscript of the "kept-constant" variables.

To determine the next lower order correlation coefficient, say, $a_{13.4567}$ in the right hand side of equation (15), we, therefore use the relationship

$$a_{13.4567} = \frac{a_{13.567} - a_{14.567}a_{34.567}}{\sqrt{(1 - a^2_{14.567})(1 - a^2_{34.567})}} \tag{16}$$

in which X_4 is now the mediator. Ultimately, these coefficients are related to the zero order correlation coefficients a_{12}, a_{13}, ..., a_{23}, a_{24}, ..., a_{67} where

$$a_{ij} = \frac{N\sum X_iX_j - (\sum X_i)(\sum X_j)}{\sqrt{[N\sum X_i^2 - (\sum X_i)^2][N\sum X_j^2 - (\sum X_j)^2]}} \tag{17}$$

In the program, the determination of the coefficients of partial correlation is carried out by the subroutine `PartCorr`. The sums $\sum X_i$, $\sum X_i^2$, and $\sum X_iX_j$ are mapped into the arrays `sumX(i)`, `sumX2(i)`, and `sumXX(i)` respectively. The correlation coefficients are denoted by the 3-D array `corr(m,i,j)` in which `(m-1)` is the order of the correlation, `i` is the subscript of the dependent variable X_i and `j` is that of the independent variable X_j. The values of the correlation that we are looking for are therefore stored in `corr(6,1,k)`, `k` $= 2, \ldots, 7$.

The zero order correlation coefficients are evaluated first and are determined by the statements

```
do m1 = 1,6
  do m2 = (m1+1),7
      corr(1,m1,m2) = (N*sumXX(m1,m2)-sumX(m1)*sumX(m2))
*         /sqrt((N*sumX2(m1)-sumX(m1)*sumX(m1))*
*               (N*sumX2(m2)-sumX(m2)*sumX(m2)))
  end do
end do
```

The mediator for the determination of each order of the correlation coefficients is preselected in the subroutine `Rank`. The "ranking" of the mediator subscript is stored in the array `mediator(m)` so that the highest order (i.e. `m` = 6) would have the lowest subscript in the range of 2 to 7 excluding the value of `k` when determining `corr(6,1,k)`. The next highest order would then have the next lowest subscript and so on. For example, to determine the coefficient $a_{15.23467}$, the "ranking" of the mediator subscripts is

```
order    :  2  3  4  5  6
mediator :  7  6  4  3  2
```

To calculate the 1st order correlation coefficients, the variable with the subscript 7 is to be the mediator. Hence the coefficients $a_{12.7}$, $a_{13.7}$, $a_{14.7}$, ..., $a_{56.7}$ are evaluated.

Because of the symmetrical relationship that `corr(m,i,j) = corr(m,j,i)`, only the values of `corr(m,i,j)` in which `i` < `j` are evaluated and used in the calculations.

5 Program Listing

The following pages are a full listing of the program `SGBCalib`. The program is intended to be executed interactively on a computer terminal. It may also be submitted as batch job in most system without any modification.

The program listing is roughly divided into 3 sections:

Main section This section consists of the main routine `SGBCalib` and the subroutines `SumElements` and `Hfactors`. They are the routines that implement the LS method.

Statistical section This includes the subroutines `Statistics`, `StatsInitialize`, `SumStats`, `SGBReading`, `PartCorr`, `Rank`, `Chart` and `Resetline` and are responsible for the evaluations of the statistical variables.

Matrix Operations The matrix inversion `MatInv` and matrix multiplication `MatMult` routines are included for the sake of completion of the program listing. There are other minor subroutines following `MatInv` and are called by `MatInv` only.

```
      Program SGBCalib
!-------------------------------------------------------------------!
!                                                                   !
!   Strain Gauge Balance Calibration using Least Squares Method     !
!                                                                   !
!    Given a set of S.G.B. readings, R(i), from a set of            !
!    known loadings, H(i), the matrices [E] and [A] are formed.     !
!    The coefficient matrix, [C], is obtained according to          !
!     the equation                                                  !
!            [C] = [Einv][A]                                        !
!                                                                   !
!   Written by S. Lam - 31 March 1988                               !
!                                                                   !
!   Associated routines:- Hfactors(H)                               !
!                         SumElements(R)                            !
!                         Statistics(N)                             !
!                         Hfactors(H)                               !
!                         MatInv(dim, E, Einv, error)               !
!                         MatMult(C1,m,n,C2,n,k,C3,error)           !
!                                                                   !
!-------------------------------------------------------------------!
      double precision  E(33,33),Einv(33,33),A(33,6),C(33,6)
      double precision  H(6),R(6),Hstar(27),Rm(6)
      double precision  sumR(6),HH(6,200),RR(6,200)
      integer error, Hsubscript(33)
      common /inv/Einv  /matrixE/E  /matrixA/A  /Hvector/H,Hstar
      common /matrixC/C  /loads/HH  /readings/RR  /misc/Rm
      data Hsubscript/1,2,3,4,5,6,11,22,33,44,55,66,12,13,14,15,16,
     *     23,24,25,26,34,35,36,45,46,56,111,222,333,444,555,666/

C -------------------------------------------------------------
C   Files for I/O:-
C       SGBIN.DAT  = file containing sets of H(i) and R(i) for
C                    data input
C       SGBOUT.DAT = file for output of results
C -------------------------------------------------------------
      open(unit=3, file='SGBIN.DAT', status='old')
      open(unit=4, file='SGBOUT.DAT', status='new')
C -------------------------------------
C  Print Headings of the program outputs
C -------------------------------------
      write(4,10)
  10  format(/,'  STRAIN GAUGE BALANCE CALIBRATION USING LEAST'
     *   ' SQUARES METHOD'/' ----------------------------------'
     *   '--------------------------'//'  The Supplied Component Loads'
     *   ' and Strain Gauge Bridge Readings:-'/)
```

```
      write(4,15) (i, i=1,6), (j, j=1,6)
   15 format(6(5x,'H',i1,3x),6(5x,'R',i1,3x))
C ------------------------------------
C   Initialize the [E] and [A] matrices
C ------------------------------------
      N = 0
      do n_row = 1,33
        do  n_col = 1,33
          E(n_row,n_col) = 0.0
        end do
      end do
      do n_col = 1,6
        do  n_row = 1,33
          A(n_row,n_col) = 0.0
        end do
        sumR(i) = 0.0
      end do
C ----------------------------------------
C  Input data :- H(i) = SGB readings vector
C                R(i) = Load vector
C ----------------------------------------
   20 read(3,*,end=50)
      read(3,*,end=50) (H(i),i=1,6)
      read(3,*,end=50) (R(i),i=1,6)
      write(4,30) (H(i),i=1,6), (R(i),i=1,6)
   30 format(12(x,f9.5))
      N = N + 1
      do i = 1,6
        sumR(i) = sumR(i) + R(i)
C ---------------------------------------------------------------
C  Store the original H(i) and R(i) values for later computations
C ---------------------------------------------------------------
        HH(i,N) = H(i)
        RR(i,N) = R(i)
      end do
C ----------------------------------------------------------------
C  Evaluate the Least Squares elements for the [E] and [A] matrices
C ----------------------------------------------------------------
      call Hfactors(H)
      call SumElements(R)
      go to 20
   50 continue
      write(4,60) N
   60 format('     Number of sets of loading = ',I3,/)
```

```
C ----------------------------
C  Evaluate the inverse of [E]
C ----------------------------
      call MatInv(33, E, Einv, error)
      if (erro .gt. 0) go to 80  !{ No inverse }
C ---------------------------------------------
C  Evaluate the coefficient matrix [C] = [Einv][A]
C ---------------------------------------------
      call MatMult(Einv,33,33,A,33,6,C,error)
      write(4,70) (i, i=1,6)
  70  format(1H1,/,'     The calibration coefficient matrix [C] :-'//,
     *4x,' H''s',6(7x,'C',i1,'-',4x))
      do i = 1,33
        write(4,100) Hsubscript(i), (C(i,j), j=1,6)
      end do
      do i = 1,6
        Rm(i) = sumR(i)/float(N)    !{ mean values of R(i) }
      end do
      call Statistics(N)
      go to 110
  80  write(*,90)
  90  format(/' %%% INVERSE OF [E] DOES NOT EXIST %%%')
 100  format(4x,i3,6(2x,f12.7))
 110  close(4)
      end
```

```
      Subroutine SumElements(R)
C ------------------------------------------------
C   Evaluate the sums which make up the elements
C   in the [E] and [A] matrices
C------------------------------------------------
      double precision  A(33,6), E(33,33), Hfactor(33), R(6)
      common /matrixE/E  /matrixA/A  /inv/Einv  /Hvector/Hfactor
      integer row, col

      do row = 1,33
        do col = 1,33
          E(row,col) = E(row,col) + Hfactor(row)*Hfactor(col)
        end do
      end do
      do col = 1,6
        do row = 1,33
          A(row,col) = A(row,col) + R(col)*Hfactor(row)
        end do
      end do
      return
      end
```

```
      Subroutine Hfactors(H)
C -------------------------------------
C   Calculate the non-linear load factors
C -------------------------------------
      double precision  H(6), Hfactor(33)
      common /Hvector/Hfactor

      do i = 1,6
        Hfactor(i) = H(i)                !{ linear }
        Hfactor(i+6) = H(i)*H(i)         !{ squared }
        Hfactor(i+27) = H(i)*H(i)*H(i)   !{ cubed }
      end do
      i = 13
      do  j = 1,5
        do k = (j+1),6
          Hfactor(i) = H(j)*H(k)         !{ cross product }
          i = i+1
        end do
      end do
      return
      end
```

```
      Subroutine Statistics(N)
C -------------------------------------------------------------
C  This subroutine evaluates the various statistics relating to
C  the R(i) values as estimated by the calibration equation
C
C    Input: N = number of sets of loading
C
C    Associate routines :- SumStats(j)
C                          StatsInitialize
C                          SGBReading(C1,C2,Re,H,error)
C                          Tstats(ndf,t)
C                          PartCorr(N,i,cpc)
C                          Chart(scale,N)
C -------------------------------------------------------------
      double precision C(33,6),C1(6,6),C2(6,27),R(6),H(6),Re(6),Rm(6)
      double precision HH(6,200),RR(6,200)
      double precision sumRH(6,6),sumR(6),sumH(6),sumR2(6),sumH2(6)
      double precision sumHH(7,7),Se(6),SdR(6),SdH(6),cmc(6),c1(6,6)
      double precision cpc(6,6),dRRm2(6),dReRm2(6),dRRe2(6),res(6,200)
      common  /Sums/sumRH,sumR,sumH,sumR2,sumH2,sumHH,
     *              dRRe2,dReRm2,dRRm2
      common  /loads/HH  /readings/RR  /misc/Rm,R,Re,H
      common  /residual/res,resmax  /matrixC/C
      parameter  (nv=33)    !{ no. of variables }

      resmax = 1.0e-20         !{ maximum res (residual) for chart scaling }
      call StatsInitialize   !{ Initialize the variables to zero }
C ----------------------------------------
C  Break up the coefficient matrix into
C  linear (C1) and non-linear (C2) matrices
C ----------------------------------------
      do i = 1, 6
        do j = 1,6
          C1(i,j) = C(j,i)
        end do
        do j = 1,27
          C2(i,j) = C(j+6,i)
        end do
      end do
C -------------------------------------------------------------
C  Restore the original H(i) and R(i) values for computations
C -------------------------------------------------------------
      do j = 1, N
        do i = 1, 6
          H(i) = HH(i,j)
          R(i) = RR(i,j)
        end do
```

```
C ------------------------------------------------------------------------
C  Estimate the values of R, Re(i), from the coefficients matrices
C  C1 and C2, and evaluate the various sums for statistical calculations
C ------------------------------------------------------------------------
        call SGBReading(C1,C2,Re,H,error)
        call SumStats(j)
C ------------------------------------------------------
C  Use HH(i,j) as storage for Re(i) for later print-outs
C ------------------------------------------------------
        do i = 1, 6
          HH(i,j) = Re(i)
        end do
      end do
  20  ndf = float(N-nv)      !{ degree of freedom }
      if (N .le. nv) ndf = 1
      fN = float(N)
      call Tstats(ndf,t)     !{ find the t-statistics }
C ------------------------------------------------------------
C  Evaluate the statistics
C    Se(i) = standard error of estimates
C    SdR(i) = standard deviation of R(i)
C    SdH(i) = standard deviation of H(i)
C    cmc(i) = coefficient of multiple correlation
C    cpc(ij) = coefficient of partial correlation
C    cl(ij) = 95% confidence level of the linear coefficients
C ------------------------------------------------------------
      do i = 1,6
        Se(i) = sqrt(dRRe2(i)/ndf)
        SdR(i) = sqrt(dRRm2(i)/ndf)
        SdH(i) = sqrt((sumH2(i)-sumH(i)*sumH(i)/fN)/ndf)
        cmc(i) = sqrt(dReRm2(i)/(dRRe2(i)+dReRm2(i)))
        call PartCorr(N,i,cpc)
      end do
      do j = 1,6
        do i = 1,6
          cl(i,j) = t/sqrt(float(ndf))*Se(i)/SdH(j)
        end do
      end do
C ------------------------------------
C  Write the results to the output file
C ------------------------------------
      write(4,30)
      write(4,110) (i, i=1,6)
      write(4,60) (Se(i), i=1,6)
  30  format(1H1,///,' Standard errors of estimates: ',/)
      write(4,40)
      write(4,110) (i, i=1,6)
```

```
      write(4,60) (cmc(i), i=1,6)
   40 format(///,' Coefficients of multiple correlation:-'/)
      write(4,50)
      write(4,120) (i, i=1,6)
   50 format(///,' Coefficients of partial correlation:-'/)
      do j = 1, 6
        write(4,65) j, (cpc(i,j), i = 1,6)
      end do
   60 format(6(2x,f11.5))
   65 format(3x,'H',i1,6(2x,f11.5))
      write(4,70) (i, i=1,6)
   70 format(///    )5% confidence level of the linear coeff.:-'//
     *        2x,6 5x,'C',i1,'-',4x))
      do j = 1, 6
        write(4,80) (cl(i,j), i = 1,6)
      end do
   80 format(6(2x,f10.7))
C -------------------------------------------------------------
C  Print out the measured ,estimated and the residuals of R(i)
C -------------------------------------------------------------
      write(4,90) (i, i=1,6)
   90 format(1H1,'   Table of residuals:- ',//,
     *   6(8x,'R',i1,11x)/6(' meas.  est.   res.  '))
      do j = 1, N
        write(4,100) (RR(i,j),HH(i,j),res(i,j), i=1,6)
  100   format(5(2(x,f5.3),x,f6.4,x,'|'),2(x,f5.3),x,f6.4)
      end do
  110 format(3x,6(6x,'R',i1,5x))
  120 format(8x,6(6x,'R',i1,5x))
C --------------------------------
C  Construct the chart of residuals
C --------------------------------
      scale = 10.0/resmax
      call Chart(scale,N)
      return
      end
```

```
      Subroutine StatsInitialize
C -------------------------------------------------------------
C  Initializes the variables of the Statistics routine to zero
C -------------------------------------------------------------
      double precision  sumRH(6,6),sumR(6),sumH(6),sumR2(6),sumH2(6)
      double precision  sumHH(7,7),dRRe2(6),dRRm2(6),dReRm2(6)
      common  /Sums/sumRH,sumR,sumH,sumR2,sumH2,sumHH,
     *              dRRe2,dReRm2,dRRm2

      do i = 1, 6
        do j = 1,6
          sumRH(j,i) = 0.0
          sumHH(j+1,i+1) = 0.0
        end do
        dRRe2(i) = 0.0
        dReRm2(i) = 0.0
        dRRm2(i) = 0.0
        sumR(i) = 0.0
        sumH(i) = 0.0
        sumR2(i) = 0.0
        sumH2(i) = 0.0
      end do
      return
      end
```

```
      Subroutine SGBReading(C1,C2,R,H,error)
C -------------------------------------------------------------
C  This subroutine evaluates the six components of Strain Gauge
C  Balance Readings (R) from the six components of Loads (H).
C
C  Input:  C1,C2 - linear and non-linear coefficient matrices
C  Output: R (the SGB Reaings vector)
C  Error: if error > 0, an error has occurred
C
C  Associate routines:- Hfactors(H)
C                       MatMult(C1,m,n,C2,n,k,C3,error)
C -------------------------------------------------------------
      double precision C1(6,6),C2(6,27),X(6),R(6),H(6),Hstar(27)
      integer error
      common /Hvector/X,Hstar

      call Hfactors(H)
C ----------------------------------------
C  Perform the matrix multipications:
C       [C1][H]  and  [C2][H*]
C ----------------------------------------
      Call MatMult(C1,6,6,H,6,1,R,error)
      if (error .gt. 0) return
      Call MatMult(C2,6,27,Hstar,27,1,X,error)
      if (error .gt. 0) return
      do i = 1,6
        R(i) = R(i) + X(i)
      end do
      return
      end
```

```
      Subroutine SumStats(j)
C -------------------------------------------------
C  Evaluate the sums for the statistical estimations
C -------------------------------------------------
      double precision sumRH(6,6),sumR(6),sumH(6),sumR2(6),sumH2(6)
      double precision sumHH(7,7),res(6,200),dRRe2(6),dRRm2(6),dReRm2(6)
      double precision  H(6),R(6),Re(6),Rm(6)
      common  /Sums/sumRH,sumR,sumH,sumR2,sumH2,sumHH,
     *              dRRe2,dReRm2,dRRm2
      common  /misc/Rm,R,Re,H  /residual/res,resmax

      do i = 1,6
        res(i,j) = R(i)-Re(i)
        if (abs(res(i,j)) .gt. resmax) resmax = abs(res(i,j))
        dRRe2(i) = dRRe2(i) + res(i,j)**2
        dRRm2(i) = dRRm2(i) + (R(i)-Rm(i))**2
        dReRm2(i) = dReRm2(i) + (Re(i)-Rm(i))**2
        sumR(i) = sumR(i) + R(i)
        sumH(i) = sumH(i) + H(i)
        sumR2(i) = sumR2(i) + R(i)*R(i)
        sumH2(i) = sumH2(i) + H(i)*H(i)
        do m = 1,6
          sumRH(i,m) = sumRH(i,m) + R(i)*H(m)
          sumHH(i+1,m+1) = sumHH(i+1,m+1) + H(i)*H(m)
        end do
      end do
      return
      end
```

```
      Subroutine Tstats(ndf,t)
C -------------------------------------------------------
C  Estimate the t-statistics from the degree of freedom
C -------------------------------------------------------
      dimension  tdf(30)
      data  tdf/12.706,4.303,3.182,2.776,2.571,2.447,2.365,2.306,
     *          2.262,2.228,2.201,2.179,2.160,2.145,2.131,2.120,
     *          2.110,2.101,2.093,2.086,2.080,2.074,2.069,2.064,
     *          2.060,2.056,2.052,2.048,2.045,2.042/

      if (ndf .le. 30) then
        t = tdf(ndf)
      else if (ndf .le. 40) then
        t = tdf(30)-0.021*float((ndf-30)/10)
      else if (ndf .le. 60) then
        t = 2.021-0.021*float((ndf-40)/10)
      else if (ndf .le. 120) then
        t = 2.000-0.020*float((ndf-60)/10)
      else
        t = 1.96
      end if
      return
      end
```

```
      Subroutine PartCorr(N,iR,cpc)
C ------------------------------------------------------------------
C  Subroutine to evaluate the coefficients of partial correlation
C
C    Inputs:- N = total number of sets of data
C             iR = the subscript of R whose coeff. of part. corr.
C                  are to be evaluated
C    Outputs:- cpc = arrays containing the coeff. of part. corr.
C
C    Cautions:- Errors may occur if, during the evaluation process,
C               the values of certain correlation coefficients are
C               close to 1.
C
C    Associated routine :- Rank(m,n)
C ------------------------------------------------------------------
      double precision  sumRH(6,6),sumR(6),sumH(6),sumR2(6),sumH2(6)
      double precision  sumXX(7,7),sumX(7),sumX2(7)
      double precision  corr(6,6,7),cpc(6,6),a,b,c
      integer  mediator(6), order
      common  /Sums/sumRH,sumR,sumH,sumR2,sumH2,sumXX

C ------------------------------------------------------
C  Map the sums of R(iR) and H(i) into the array sumX(j)
C      the sums of squares into the array sumX2(j), and
C      the sums of cross products into sumXX(j,k)
C ------------------------------------------------------
      sumX(1) = sumR(iR)
      sumX2(1) = sumR2(iR)
      do j = 2,7
        sumX(j) = sumH(j-1)
        sumX2(j) = sumH2(j-1)
        sumXX(1,j) = sumRH(iR,j-1)
      end do
C ------------------------------------------------------------------
C  Evaluate the zero order (order = 1) correlation coefficients first
C ------------------------------------------------------------------
      do m1 = 1,6
        do m2 = (m1+1),7
            corr(1,m1,m2) = (N*sumXX(m1,m2)-sumX(m1)*sumX(m2))
     *          /sqrt((N*sumX2(m1)-sumX(m1)*sumX(m1))*
     *                (N*sumX2(m2)-sumX(m2)*sumX(m2)))
        end do
      end do
```

```
C ------------------------------------------------------
C  Start evaluating higher order correlation coefficients
C ------------------------------------------------------
      do i = 2,7        !{ i = component subscript }
        call rank(i,mediator)
        do order = 2,6          !{ order of correlation coefficient }
          med = mediator(order) !{ subscript of the mediator }
          do n1 = 1,(med-1)     !{ subscript of the 1st variable }
            do n2 = (n1+1),7    !{ subscript of the 2nd variable }
              if ((n2 .lt. med) .or. (n2 .eq. i)) then
                a = corr(order-1,n1,n2)
                b = corr(order-1,n1,med)
                c = corr(order-1,n2,med)
                if (n2 .gt. med) c = corr(order-1,med,n2)
C ------------------------------------------------------------
C  To prevent negative sqrt argument and also divide by zero errors
C  the correlation coefficient is set slightly less than 1 if
C  either b or c >= 1.
C ------------------------------------------------------------
                if (b .ge. 1.0) b = 0.999999
                if (c .ge. 1.0) c = 0.999999
                corr(order,n1,n2) = (a-b*c)/sqrt((1.0-b*b)*(1.0-c*c))
              end if
            end do      !{ end of n2 loop }
          end do        !{ end of n1 loop }
        end do          !{ end of order loop }
      end do            !{ end of i loop }
      do k = 1,6
         cpc(iR,k) = corr(6,1,k+1)
      end do
      return
      end
```

```
      Subroutine Rank(n,med)
C -----------------------------------------------------------------
C  Rank the subscripts of the mediator from 7 down to 2 excluding n
C -----------------------------------------------------------------
      integer  med(6)

      med(2) = 7
      if (n .eq. 7) med(2) = med(2)-1
      do i = 3,6
        med(i) = med(i-1) - 1
        if (med(i) .eq. n) med(i) = med(i) - 1
      end do
      return
      end
```

```
      Subroutine Chart(scale,N)
C -----------------------------------------------------------------
C  Construct and print the chart of the residuals of the difference
C  between the estimated and measured values of R(i)
C
C    Inputs:- scale = 10/resmax
C                N = no. of sets of data
C    Outputs:- line printer plot of the chart of residuals written
C              to logical unit number 4
C
C    Associate routine :- Resetline(line)
C -----------------------------------------------------------------
      double precision res(6,200)
      character line(131), symbol(6)
      data symbol/'*','o','#','*','o','#'/
      common  /residual/res,resmax

      write(4,5) resmax, (i, i=1,6)
   5  format(1H1,'  Deviations of the measured R''s from'
     *    ' the estimated R''s'/,'    Maximum deviation = ',e12.5,//,
     *   6(10x,'R',i1,10x))
      do i = 1,N
        call resetline(line)   !{ clear the line of characters }
        do j = 1,6
          index = res(j,i)*scale
          index = index + (j-1)*22 + 11
          line(index) = symbol(j)
        end do
        write(4,15) (line(k), k=1,131)
  15    format(131a1)
      end do
      return
      end
```

```
      Subroutine Resetline(line)
C -----------------------------------------------
C  Clears a line of character to contain spaces and
C  the zero marker '!' only
C  - Called by the subroutine Chart
C -----------------------------------------------
      character line(131)

      do i=1, 131
        line(i) = ' '
      end do
      do i = 11,131,22
        line(i) = '!'
      end do
      return
      end
```

```
      Subroutine MatMult(m1,row1,col1,m2,row2,col2,product,error)
C ------------------------------------------------------------------!
C  Matrix multipication routine                                      !
C                                                                    !
C  Input: m1, m2 (the first and second matrices)                     !
C         row1, col1, row2, col2 (dimensions of the two matrices)    !
C  Output: product, error                                           !
C                                                                    !
C  Error: 0 - successful operation                                  !
C         1 - the column of the first matrix does not match that of  !
C             the second                                             !
C ------------------------------------------------------------------!
      integer  row1, row2, col1, col2, error
      double precision m1(row1,col1),m2(row2,col2),product(row1,col2)

      if (col1 - row2) 10, 20, 10
C-------------------------------------------------------------------
C   No. of columns in m1 does not match that in m2 - no operation
C-------------------------------------------------------------------
  10  error = 1
      return
C -----------------------------------
C   Perform the Matrix Multipications
C -----------------------------------
  20  do i = 1, row1
        do j = 1, col2
          product(i,j) = 0.0
          do k = 1, col1
            product(i,j) = product(i,j) + m1(i,k)*m2(k,j)
          end do
        end do
      end do
      error = 0
      return
      end
```

```
      Subroutine MatInv(dimen, matrix, inverse, error)
C ------------------------------------------------------------!
C        Purpose : To calculate the inverse of a matrix      !
C                  with Gauss-Jordan elimination.            !
C                                                            !
C          Input : dimen, matrix                             !
C         Output : inverse, error                            !
C                                                            !
C        Remarks : The original matrix will be changed to    !
C                  an identity matrix.                       !
C                                                            !
C     Error code : 0 - no error                              !
C                  1 - dimension of matrix < 1               !
C                  2 - no inverse exists                     !
C                                                            !
C     Written by : S .Lam                                    !
C           Date : 2 March 1988                              !
C ------------------------------------------------------------!
      integer  dimen, error, row, ref_row, term
      double precision  matrix(dimen,dimen),inverse(dimen,dimen)
      double precision  almostzero/1.0e-20/, multiplier
      common  /zero/almostzero

      call Initial(dimen, matrix, inverse, error)
      if (dimen .le. 1) return
      ref_row = 0
      do while ((error .eq. 0) .and. (ref_row .lt. dimen))
        ref_row = ref_row + 1
C ---------------------------------------------
C   Check to see if the diagonal element is zero
C ---------------------------------------------
        if (abs(matrix(ref_row,ref_row)) .lt. almostzero)
     +    call Pivot(dimen, ref_row, matrix, inverse, error)
        if (error) 30, 20, 30
C ---------------------------------------------
C   If error = 0 , carry out the inverse operation
C ---------------------------------------------
  20    divisor = matrix(ref_row,ref_row)
        do term = 1, dimen
          matrix(ref_row,term) = matrix(ref_row,term)/divisor
          inverse(ref_row,term) = inverse(ref_row,term)/divisor
        end do
```

```
          do row = 1, dimen
C ------------------------------------------
C   Make the ref_row element of this row zero
C ------------------------------------------
            if ((row .ne. ref_row) .and.
     +         (abs(matrix(row,ref_row)) .gt. a'mostzero)) then
              multiplier = -matrix(row,ref_row)/matrix(ref_row,ref_row)
              do term = 1, dimen
                matrix(row,term) = matrix(row,term) + multiplier*
     +                           matrix(ref_row,term)
                inverse(row,term) = inverse(row,term) + multiplier*
     +                           inverse(ref_row,term)
              end do
            end if
          end do
C ---------------------------------------
C     If error <> 0, terminate operation
C ---------------------------------------
  30    end do  !{ end of do while loop }
        return
        end
```

```
      Subroutine Initial(dimen, matrix, inverse, error)
C ------------------------------------------------------------!
C   This subroutine tests for errors in the value of dimen !
C                                                            !
C   Input: dimen, matrix                                     !
C   Output: inverse, error                                   !
C ------------------------------------------------------------!
      integer  dimen, error, row, col
      double precision  matrix(dimen,dimen), inverse(dimen,dimen)
      double precision  almostzero
      common  /zero/almostzero

      error = 0
      if (dimen - 1) 10, 20, 30
C -------------------------
C     Case dimen < 1, error
C -------------------------
  10  error = 1
      return
C ---------------------------------------
C     Case dimen = 1, single element matrix
C ---------------------------------------
  20  continue
      if (abs(matrix(1,1)) .lt. almostzero) then
        error = 2    !{ singular matrix }
      else
        inverse(1,1) = 1.0/matrix(1,1)
      end if
      return
C ------------------------------------------------------------
C     Case dimen > 1, make the inverse-to-be an identity matrix
C ------------------------------------------------------------
  30  do col = 1, dimen
        do row = 1, dimen
          if (col .eq. row) then
            inverse(row, col) = 1.0
          else
            inverse(row, col) = 0.0
          end if
        end do
      end do
      return
      end
```

```
      Subroutine Pivot(dimen, ref_row, matrix, inverse, error)
C ------------------------------------------------------------------------!
C   This subroutine searches the ref_row column of the given matrix      !
C   for the first non-zero element below the diagonal.  If it finds      !
C   one, then the subroutine switches rows so that the non-zero element  !
C   is on the diagonal.  This same operation is applied to the inverse   !
C   matrix.  If no non-zero element exists in a column, the matrix is    !
C   singular and no inverse exists.                                      !
C                                                                        !
C   Input: dimen, ref_row, matrix, inverse                               !
C   Output : matrix, inverse, error                                      !
C ------------------------------------------------------------------------!
      integer  dimen, error, new_row
      double precision  matrix(dimen,dimen), inverse(dimen,dimen)
      double precision  almostzero
      common  /zero/almostzero

      error = 2   !{  No inverse exists }
      new_row = ref_row
C ---------------------------------------------------
C   Try to find a row with a non-zero diagonal element
C ---------------------------------------------------
      do while ((error .gt. 0) .and. (new_row .lt. dimen))
        new_row = new_row + 1
        if (abs(matrix(new_row,ref_row)) .gt. almostzero) then
C ------------------------------------------------
C   Switch rows between the new_row and the ref_row
C ------------------------------------------------
          call SwitchRow(matrix, dimen, new_row, ref_row)
          call SwitchRow(inverse, dimen, new_row, ref_row)
          error = 0
        end if
      end do  !{ While }
      return
      end
```

```
      Subroutine SwitchRow(matrix, dimen, row1, row2)
C ------------------------------------------------------------!
C   This subroutine switches the rows specified by row1 and row2 !
C   of the given matrix.                                       !
C                                                              !
C   Input: matrix, dimen, row1, row2                           !
C   Output: matrix                                             !
C ------------------------------------------------------------!
      integer  dimen, row1, row2, term
      double precision  matrix(dimen,dimen), dummy

      do  term = 1, dimen
        dummy = matrix(row1,term)
        matrix(row1,term) = matrix(row2,term)
        matrix(row2,term) = dummy
      end do
      return
      end
```

6 Example

As a demonstration of the functions of the program SGBCalib, a "numerical" calibration was carried out using an "artificial" balance whose calibration coefficient matrix is given as follows:

$$[C] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} . \qquad (18)$$

This represents a perfectly linear balance with no interactions between the components. It is an ideal balance which, of course, is practically impossible to attain. Nevertheless, it serves the purpose of a simple illustration of the capabilities of the calibration program.

A number of sets of randomly generated component loads H_i, (ranged from 0 to 2 units) were used, in conjunction with the coefficient matrix $[C]$ of equation (18), to produce a number of corresponding sets of strain gauge balance outputs R_i according to the relationship described by equation (2). The resulting R_i values were multiplied by a random factor in the range of ± 0.002 to simulate a random noise level of $\pm 0.2\%$ on the strain gauge balance outputs. The sets of H_i and noise inflicted R_i values were written to the input file `SGBIN.DAT` according to the format described in Section 4.2.1. The program was then executed to evaluate the elements of $[C]$ from these sets of H_i and R_i values. The outputs of the program are shown in page 39 through to page 43.

Examination of the results shows that the maximum deviation of the estimated calibration coefficients is approximately 1% from the "true" values. This is hardly acceptable in the high precision work that a strain gauge balance is expected to operate. The inaccurate estimation of the coefficients in this particular case is because not all of the randomly applied component loads necessarily give useful independent information for the evaluation of the coefficients. The $\pm 0.2\%$ random noise level further distorts this information from accurately determining the coefficients. The accuracy, however, could be improved if a larger number of sets of loadings is used. Pages 44 and 45 are an abstract of the outputs from a similar test run except that 100 sets of random loadings were used for the "calibration". The results showed considerable improvement in the accuracies of estimation of the calibration coefficients.

On the other hand, the correlation coefficients show perfect multiple correlations between the strain gauge balance outputs and the six component loads. There is also perfect linear relationship between each of the R_i and its corresponding H_i. These are to be expected because the R_i values were derived from exactly the same relationship as the regression equation.

STRAIN GAUGE BALANCE CALIBRATION USING LEAST SQUARES METHOD

The Supplied Component Loads and Strain Gauge Bridge Readings:-

H1	H2	H3	H4	H5	H6	R1	R2	R3	R4	R5	R6
0.96200	0.41700	0.19200	0.60100	0.67800	0.52100	0.96190	0.41675	0.19266	0.60042	0.67759	0.52037
1.08900	1.22900	0.65900	1.82600	0.80200	1.52000	1.08884	1.22856	0.65775	1.82401	0.80015	1.51700
0.47900	0.43700	1.20500	0.48300	0.02900	0.70200	0.47977	0.43807	1.20736	0.48352	0.02871	0.70307
1.59600	1.26200	0.47900	1.23100	1.26500	1.78400	1.59715	1.26049	0.47973	1.23269	1.26251	1.78552
0.49400	0.43300	1.00600	1.24400	1.60400	1.07900	0.49355	0.43278	1.00656	1.24510	1.60494	1.07817
1.52800	1.73300	0.18000	0.85900	0.39200	1.01100	1.52590	1.73139	0.18016	0.85822	0.39125	1.01006
0.34200	0.30500	1.71000	0.22500	0.78600	0.24000	0.34278	0.30515	1.71057	0.22554	0.78442	0.23967
1.29100	1.74100	1.91500	1.21800	1.46900	0.90500	1.29217	1.74138	1.91164	1.21768	1.46818	0.90493
0.95600	1.49400	0.78800	1.40700	0.67700	1.03800	0.95518	1.49628	0.78946	1.40616	0.67825	1.04012
0.91200	1.63100	1.23100	0.36200	0.18700	1.67000	0.91325	1.63209	1.23299	0.36176	0.18779	1.66809
0.38800	1.48100	1.36000	0.06800	0.43500	1.65500	0.38844	1.47874	1.36157	0.06763	0.43525	1.65425
1.00100	0.18000	0.30300	0.74100	1.40900	0.76200	1.00138	0.18003	0.30285	0.74085	1.41022	0.76220
1.19900	0.67100	1.37500	0.82100	1.47800	1.04300	1.20159	0.66996	1.37473	0.82095	1.48042	1.04436
1.58000	0.16900	0.05400	1.42200	1.81400	0.02200	1.57909	0.16905	0.05436	1.42079	1.81078	0.02196
1.73700	1.64200	1.66000	1.02400	1.56400	0.74100	1.73425	1.64128	1.66103	1.02417	1.56381	0.74028
0.79300	0.31900	1.13700	1.68800	0.41900	1.17400	0.79457	0.31962	1.13614	1.68933	0.41951	1.17616
1.27400	0.77600	0.84900	1.56600	0.75400	1.06800	1.27573	0.77626	0.84690	1.56839	0.75463	1.06900
1.48300	1.61300	1.80000	0.00800	0.97300	1.71000	1.48433	1.61453	1.79676	0.00806	0.97261	1.71032
1.02800	0.26600	1.17700	1.02300	1.23900	1.68700	1.02978	0.26591	1.17725	1.02117	1.23851	1.68528
1.65500	1.98200	0.62600	0.38900	0.22400	1.13800	1.65695	1.97924	0.62599	0.38972	0.22347	1.13807
1.05800	0.48700	1.16500	1.19700	1.04900	0.54900	1.05692	0.48734	1.16598	1.19601	1.04985	0.54956
0.80200	1.37600	0.14500	0.09500	0.06700	1.61600	0.80144	1.37593	0.14477	0.09532	0.06666	1.61763
1.03300	0.76000	1.16200	0.02700	1.63600	1.20600	1.03166	0.76043	1.16351	0.02740	1.63327	1.20438
0.11800	1.20200	1.32200	0.74600	1.56100	0.95200	0.11833	1.20445	1.32307	0.74437	1.56264	0.95291
0.87900	0.47100	1.77700	1.87000	1.32800	0.50500	0.88079	0.47176	1.78012	1.87119	1.32982	0.50578
0.57600	0.14200	0.92200	0.78900	0.82500	1.98600	0.57710	0.14266	0.92259	0.78776	0.82359	1.98260
1.29300	0.58300	1.17700	1.16500	0.66400	0.65500	1.29298	0.58330	1.17892	1.16404	0.66507	0.65450
0.88500	0.34600	1.74500	1.34900	0.80600	0.34300	0.88597	0.34636	1.74716	1.34829	0.80666	0.34357
0.14300	1.51000	1.26300	1.46400	1.60000	1.66600	0.14265	1.51233	1.26448	1.46517	1.60085	1.66841
1.32600	1.99300	1.74300	1.45000	0.92700	0.56000	1.32498	1.99523	1.73961	1.45028	0.92828	0.55947
0.73400	1.37200	0.72000	1.84800	0.58900	0.16500	0.73314	1.37271	0.72009	1.84710	0.58858	0.16514
1.45300	0.23900	0.00400	0.61100	0.60000	0.59100	1.45303	0.23961	0.00445	0.61019	0.59977	0.59117
1.95400	0.62600	0.94600	0.70800	1.54200	0.59200	1.95780	0.62738	0.94648	0.70756	1.54283	0.59246
1.68100	0.53000	1.62200	0.82100	0.62900	0.68900	1.68424	0.53120	1.62021	0.82155	0.62903	0.68813
0.29600	1.06700	0.84900	0.90400	0.65700	0.76200	0.29580	1.06783	0.84825	0.90552	0.65652	0.76199
0.73900	0.29300	1.42800	1.88600	0.47200	0.65400	0.73731	0.29247	1.42835	1.88479	0.47255	0.65348
1.74600	1.20900	0.52500	0.30500	1.12800	1.80600	1.74494	1.20995	0.52649	0.30474	1.12901	1.80935
1.00200	1.00300	0.60600	0.15000	1.38900	1.46600	1.00375	1.00158	0.60565	0.15047	1.38692	1.46379
1.56000	0.03600	1.80900	1.69400	1.72500	0.49200	1.55998	0.03600	1.80842	1.69165	1.72437	0.49194
1.70300	0.62500	0.18400	0.59700	1.46300	1.29000	1.70056	0.62533	0.18449	0.59743	1.46596	1.28894
1.90200	1.13500	1.69700	0.05900	1.25200	1.91200	1.90331	1.13594	1.69388	0.05869	1.25048	1.91518
0.00900	0.43100	1.39400	0.31600	0.49100	1.85100	0.00867	0.43000	1.39444	0.31608	0.49073	1.85046
1.36800	1.19800	0.65200	1.24500	0.88000	0.55100	1.36778	1.19943	0.65173	1.24423	0.88197	0.55174
1.97900	0.18900	0.90900	1.35900	0.75500	1.18900	1.97900	0.18847	0.90958	1.36194	0.75536	1.18878
0.38000	0.89900	1.77900	1.99700	0.35300	1.72600	0.37984	0.89953	1.77862	1.99560	0.35295	1.72450

Number of sets of loading = 45

The calibration coefficient matrix [C] :-

H's	C1-	C2-	C3-	C4-	C5-	C6-
0	1.0012351	-0.0063413	0.0154363	0.0014695	-0.0059688	-0.0067772
0	0.0129323	0.9913341	-0.0143336	0.0061434	-0.0123949	-0.0027020
3	0.0109221	0.0038757	0.9988513	0.0043933	-0.0035543	0.0044627
4	0.0132318	-0.0105562	-0.0042239	0.9907176	0.0005138	-0.0121343
5	-0.0145379	-0.0007988	0.0037700	0.0028336	0.9951605	-0.0027273
6	-0.0139431	0.0095085	-0.0027626	0.0014356	0.0027766	1.0106157
11	0.0007232	0.0129578	-0.0138237	-0.0040022	0.0082517	0.0098461
22	-0.0148358	0.0101946	0.0124597	-0.0085651	0.0135931	-0.0007856
33	-0.0166860	0.0002681	0.0086865	-0.0050335	0.0077215	-0.0003932
44	-0.0107682	0.0119613	-0.0027164	0.0076425	0.0015101	0.0160626
55	0.0178118	0.0049139	-0.0054122	-0.0058849	0.0119856	0.0038632
66	0.0106750	-0.0082985	0.0014104	0.0015134	0.0041594	-0.0101144
12	-0.0013178	-0.0024763	0.0022963	0.0001596	0.0013849	-0.0029943
13	0.0014860	-0.0001528	-0.0035659	0.0000886	-0.0016691	-0.0013002
14	0.0012542	-0.0047644	0.0008007	-0.0002157	-0.0026434	-0.0050592
15	-0.0006646	-0.0004316	-0.0010192	-0.0007182	0.0003101	0.0017703
16	-0.0026408	0.0024906	0.0001276	0.0012296	-0.0005563	0.0049177
23	0.0008008	0.0001710	-0.0007229	0.0001122	0.0024688	-0.0038721
24	-0.0020259	0.0022987	0.0018824	0.0002890	0.0010410	0.0024124
25	-0.0008875	0.0012775	0.0005564	0.0015410	-0.0031737	0.0052127
26	0.0021239	-0.0019674	0.0015035	-0.0007348	-0.0000437	0.0002200
34	-0.0017599	0.0002677	0.0032829	0.0002858	0.0017971	0.0006358
35	0.0024649	-0.0014868	-0.0005776	-0.0011839	-0.0001067	-0.0007221
36	0.0001083	-0.0006518	0.0012337	-0.0001878	-0.0010117	0.0020705
45	-0.0019059	0.0015455	0.0021212	0.0025547	0.0012142	0.0029439
46	0.0008126	0.0006086	-0.0013780	-0.0001456	-0.0015151	-0.0011441
56	0.0015409	-0.0044099	0.0002235	-0.0005195	0.0001183	-0.0066371
111	-0.0002125	-0.0042487	0.0041602	0.0016827	-0.0020479	-0.0032262
222	0.0047494	-0.0031102	-0.0045295	0.0026395	-0.0049991	0.0012629
333	0.0056439	-0.0006310	-0.0041301	0.0017161	-0.0032534	-0.0004954
444	0.0031460	-0.0037433	0.0007379	-0.0023590	-0.0007855	-0.0054015
555	-0.0066238	-0.0016760	0.0019105	0.0022104	-0.0056942	-0.0018099
666	-0.0025328	0.0025933	-0.0005954	-0.0010584	-0.0025058	0.0026008

Standard errors of estimates:-

R1	R2	R3	R4	R5	R6
0.00164	0.00107	0.00135	0.00134	0.00123	0.00133

Coefficients of multiple correlation:-

R1	R2	R3	R4	R5	R6
1.00000	1.00000	1.00000	1.00000	1.00000	1.00000

Coefficients of partial correlation:-

	R1	R2	R3	R4	R5	R6
H1	1.00000	-0.14287	-0.32993	0.07504	-0.02823	0.08952
H2	-0.25572	1.00000	-0.20926	0.01755	0.05280	0.13754
H3	0.33395	0.17363	1.00000	-0.00733	0.08608	0.01009
H4	-0.16915	0.08317	-0.07035	1.00000	0.13845	-0.00929
H5	-0.11658	0.15475	0.01670	-0.06337	1.00000	0.05361
H6	0.11697	-0.13357	-0.10972	0.07715	-0.08667	1.00000

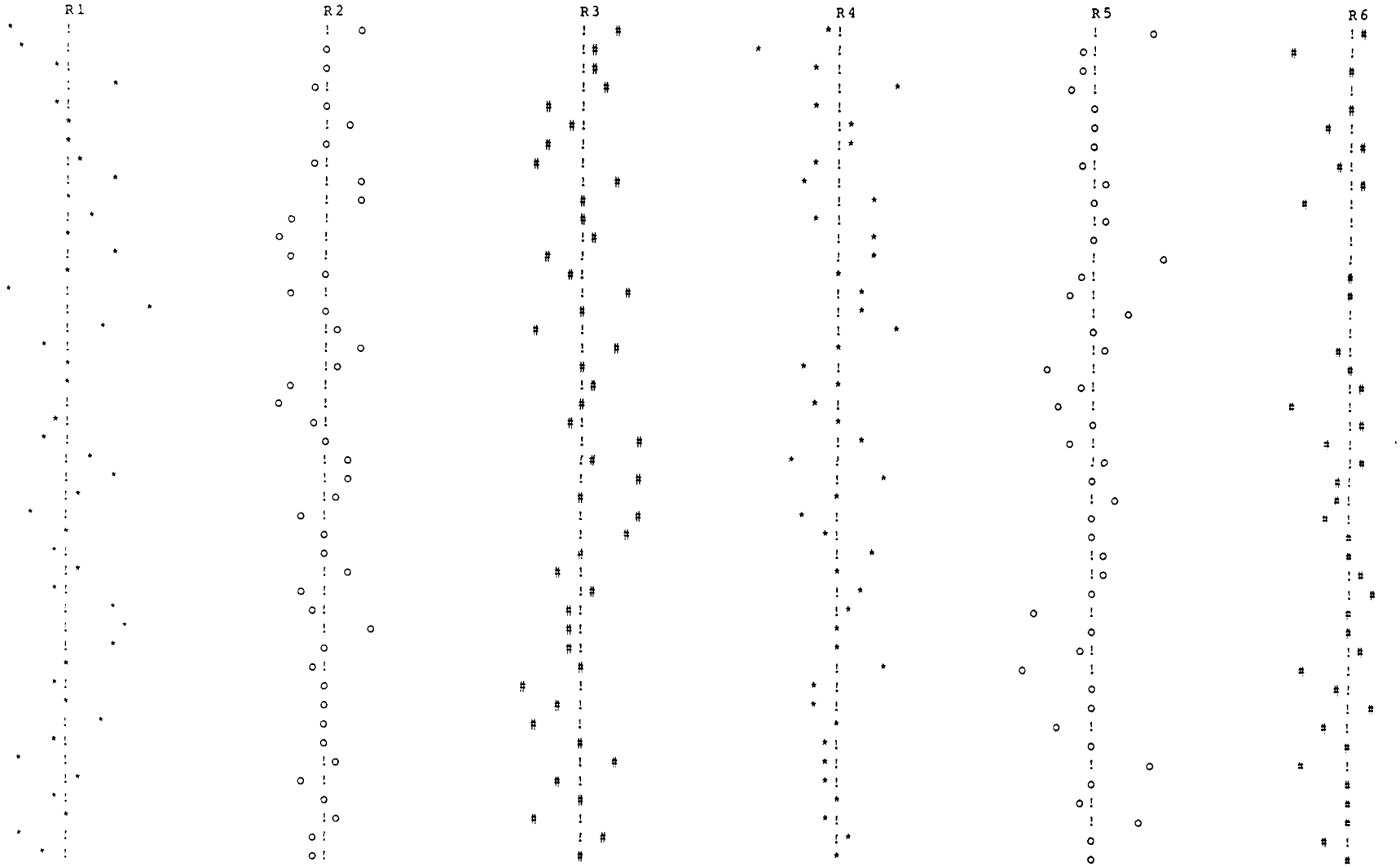
95% confidence level of the linear coeff.:-

C1-	C2-	C3-	C4-	C5-	C6-
0.0010162	0.0006635	0.0008364	0.0008300	0.0007651	0.0008256
0.0009447	0.0006168	0.0007775	0.0007716	0.0007112	0.0007675
0.0009719	0.0006345	0.0007999	0.0007939	0.0007317	0.0007896
0.0009191	0.0006001	0.0007565	0.0007507	0.0006920	0.0007467
0.0010919	0.0007129	0.0008987	0.0008919	0.0008221	0.0008872
0.0010073	0.0006576	0.0008290	0.0008228	0.0007583	0.0008184

Table of residuals:-

R1			R2			R3			R4			R5			R6		
meas.	est.	res.	meas.	est.	res.	meas.	est.	res.	meas.	est.	res.	meas.	est.	res.	meas.	est.	res
0.962	0.963	-.0014	0.417	0.416	0.0008	0.193	0.192	0.0008	0.600	0.601	-.0004	0.678	0.676	0.0014	0.520	0.520	0.000
1.089	1.090	-.0010	1.229	1.228	0.0001	0.658	0.657	0.0003	1.824	1.826	-.0019	0.800	0.801	-.0004	1.517	1.518	-.001
0.480	0.480	-.0005	0.438	0.438	0.0002	1.207	1.207	0.0003	0.484	0.484	-.0005	0.029	0.029	-.0003	0.703	0.703	-.000
1.597	1.596	0.0011	1.260	1.261	-.0004	0.480	0.479	0.0006	1.233	1.231	0.0013	1.263	1.263	-.0005	1.786	1.785	0.001
0.494	0.494	-.0003	0.433	0.433	0.0001	1.007	1.007	-.0008	1.245	1.246	-.0006	1.605	1.605	-.0001	1.078	1.078	0.000
1.526	1.526	-.0002	1.731	1.731	0.0006	0.180	0.180	-.0003	0.858	0.858	0.0003	0.391	0.391	0.0000	1.010	1.011	-.000
0.343	0.343	0.0001	0.305	0.305	-.0002	1.711	1.711	-.0009	0.226	0.225	0.0004	0.784	0.784	0.0002	0.240	0.239	0.000
1.292	1.292	0.0004	1.741	1.742	-.0004	1.912	1.913	-.0010	1.218	1.218	-.0006	1.468	1.469	-.0004	0.905	0.905	-.000
0.955	0.954	0.0011	1.496	1.495	0.0008	0.789	0.789	0.0009	1.406	1.407	-.0007	0.678	0.678	0.0003	1.040	1.040	0.000
0.913	0.913	-.0001	1.632	1.631	0.0009	1.233	1.233	0.0000	0.362	0.361	0.0010	0.188	0.188	0.0001	1.668	1.669	-.001
0.388	0.388	0.0006	1.479	1.479	-.0007	1.362	1.362	-.0001	0.068	0.068	-.0006	0.435	0.435	0.0004	1.654	1.653	0.001
1.001	1.001	0.0000	0.180	0.181	-.0011	0.303	0.303	0.0003	0.741	0.740	0.0009	1.410	1.410	0.0000	0.762	0.761	0.001
1.202	1.201	0.0010	0.670	0.671	-.0007	1.375	1.376	-.0009	0.821	0.820	0.0009	1.480	1.479	0.0015	1.044	1.043	0.001
1.579	1.579	0.0002	0.169	0.169	0.0000	0.054	0.055	-.0002	1.421	1.421	-.0002	1.811	1.811	-.0003	0.022	0.022	-.000
1.734	1.736	-.0013	1.641	1.642	-.0008	1.661	1.660	0.0012	1.024	1.024	0.0006	1.564	1.564	-.0005	0.740	0.740	0.000
0.795	0.793	0.0019	0.320	0.319	0.0002	1.136	1.136	-.0002	1.689	1.689	0.0007	0.420	0.419	0.0008	1.176	1.175	0.001
1.276	1.275	0.0009	0.776	0.776	0.0003	0.847	0.848	-.0011	1.568	1.567	0.0014	0.755	0.754	0.0002	1.069	1.068	0.001
1.484	1.485	-.0007	1.615	1.614	0.0009	1.797	1.796	0.0007	0.008	0.008	0.0002	0.973	0.972	0.0003	1.710	1.711	-.000
1.030	1.030	-.0001	0.266	0.266	0.0004	1.177	1.177	-.0001	1.021	1.022	-.0009	1.239	1.240	-.0011	1.685	1.685	-.000
1.657	1.657	0.0002	1.979	1.980	-.0009	0.626	0.626	0.0005	0.390	0.390	-.0002	0.223	0.224	-.0003	1.138	1.138	0.000
1.057	1.059	-.0024	0.487	0.488	-.0010	1.166	1.166	-.0002	1.196	1.197	-.0005	1.050	1.051	-.0008	0.550	0.551	-.001
0.801	0.802	-.0004	1.376	1.376	-.0003	0.145	0.145	-.0004	0.095	0.095	-.0001	0.067	0.067	0.0001	1.618	1.617	0.000
1.032	1.032	-.0007	0.760	0.760	0.0001	1.164	1.162	0.0012	0.027	0.027	0.0006	1.633	1.634	-.0005	1.204	1.205	-.000
0.118	0.118	0.0005	1.204	1.204	0.0007	1.323	1.323	0.0003	0.744	0.746	-.0012	1.563	1.562	0.0004	0.953	0.953	0.000
0.881	0.880	0.0011	0.472	0.471	0.0006	1.780	1.779	0.0013	1.871	1.870	0.0010	1.330	1.330	-.0002	0.506	0.506	-.000
0.577	0.577	0.0004	0.143	0.142	0.0004	0.923	0.922	0.0002	0.788	0.788	0.0001	0.824	0.823	0.0007	1.983	1.983	-.000
1.293	1.294	-.0009	0.583	0.584	-.0006	1.179	1.178	0.0013	1.164	1.165	-.0010	0.665	0.665	0.0002	0.654	0.655	-.000
0.886	0.886	-.0002	0.346	0.346	0.0001	1.747	1.746	0.0011	1.348	1.349	-.0003	0.807	0.807	0.0000	0.344	0.344	0.000
0.143	0.143	-.0005	1.512	1.512	-.0001	1.264	1.264	0.0000	1.465	1.464	0.0008	1.601	1.600	0.0004	1.668	1.668	0.000
1.325	1.325	0.0003	1.995	1.995	0.0006	1.740	1.740	-.0007	1.450	1.450	-.0001	0.928	0.928	0.0003	0.559	0.559	0.000
0.733	0.733	-.0003	1.373	1.373	-.0006	0.720	0.720	0.0003	1.847	1.847	0.0006	0.589	0.589	0.0000	0.165	0.164	0.000
1.453	1.452	0.0012	0.240	0.240	-.0005	0.004	0.005	-.0005	0.610	0.610	0.0003	0.600	0.601	-.0014	0.591	0.591	0.000
1.958	1.956	0.0014	0.627	0.626	0.0012	0.946	0.947	-.0004	0.708	0.708	-.0002	1.543	1.543	0.0002	0.592	0.592	0.000
1.684	1.683	0.0010	0.531	0.531	0.0001	1.620	1.621	-.0004	0.822	0.821	0.0002	0.629	0.629	-.0002	0.688	0.688	0.000
0.296	0.296	0.0001	1.068	1.068	-.0003	0.848	0.848	-.0001	0.906	0.904	0.0011	0.657	0.658	-.0017	0.762	0.763	-.001
0.737	0.738	-.0003	0.292	0.292	0.0001	1.428	1.430	-.0012	1.885	1.885	-.0005	0.473	0.473	0.0000	0.653	0.654	-.000
1.745	1.745	-.0002	1.210	1.210	0.0001	0.526	0.527	-.0007	0.305	0.305	-.0007	1.129	1.129	0.0002	1.809	1.809	0.000
1.004	1.003	0.0008	1.002	1.002	-.0001	0.606	0.607	-.0010	0.150	0.150	0.0001	1.387	1.388	-.0009	1.464	1.464	-.000
1.560	1.560	-.0004	0.036	0.036	0.0001	1.808	1.809	-.0002	1.692	1.692	-.0003	1.724	1.724	0.0002	0.492	0.492	0.000
1.701	1.702	-.0012	0.625	0.625	0.0004	0.184	0.184	0.0009	0.597	0.598	-.0004	1.466	1.465	0.0012	1.289	1.290	-.001
1.903	1.903	0.0004	1.136	1.136	-.0005	1.694	1.694	-.0006	0.059	0.059	-.0004	1.250	1.251	-.0001	1.915	1.915	0.000
0.009	0.009	-.0004	0.430	0.430	-.0002	1.394	1.394	0.0002	0.316	0.316	0.0002	0.491	0.491	-.0003	1.850	1.851	-.000
1.368	1.368	0.0000	1.199	1.199	0.0003	0.652	0.653	-.0010	1.244	1.245	-.0003	0.882	0.881	0.0011	0.552	0.552	0.000
1.979	1.980	-.0012	0.188	0.189	-.0003	0.910	0.909	0.0005	1.362	1.362	0.0004	0.755	0.755	0.0000	1.189	1.189	-.000
0.380	0.380	-.0006	0.900	0.900	-.0004	1.779	1.779	0.0001	1.996	1.996	0.0000	0.353	0.353	-.0002	1.725	1.724	0.000

Deviations of the measured R's from the estimated R's
Maximum deviation = 0.24377E-02



An abstract of the outputs of a test case using 100 random sets of loadings

---

The calibration coefficient matrix [C] :-

H's	C1-	C2-	C3-	C4-	C5-	C6-
0	1.0001717	0.0009336	-0.0004295	-0.0007010	-0.0001990	0.0002503
0	0.0004810	0.9999368	0.0002292	0.0001573	-0.0001607	0.0008645
3	0.0006427	-0.0001959	0.9993358	-0.0005989	0.0000927	-0.0010139
4	-0.0009255	-0.0003814	-0.0008730	0.9998394	0.0001361	-0.0004981
5	0.0001514	0.0008537	0.0008604	0.0009582	0.9999268	-0.0002699
6	-0.0008279	-0.0009503	0.0002611	-0.0014950	-0.0002893	1.0008528
11	-0.0006976	-0.0015109	0.0002755	0.0006690	0.0003722	-0.0001347
22	-0.0006716	0.0005514	-0.0006500	-0.0002395	0.0000489	-0.0006435
33	-0.0007450	0.0000834	0.0014033	0.0012474	-0.0000053	0.0007644
44	0.0011022	0.0005609	0.0007541	0.0003915	0.0002252	0.0002331
55	0.0000946	-0.0010105	-0.0009485	-0.0005102	-0.0002841	0.0003385
66	0.0009363	0.0006462	0.0000283	0.0016981	0.0007122	-0.0009687
12	0.0001438	0.0001278	0.0000281	-0.0000382	0.0000784	0.0001262
13	-0.0000892	-0.0001011	-0.0003451	-0.0001032	0.0001324	0.0000745
14	-0.0000256	-0.0001501	0.0002764	0.0000007	-0.0002960	-0.0001746
15	0.0001167	0.0002010	-0.0000922	0.0000914	0.0001090	0.0001298
16	0.0000337	0.0001325	0.0000419	0.0000316	-0.0001244	-0.0001752
23	0.0000520	0.0000986	0.0001432	-0.0000547	0.0000270	0.0000877
24	0.0000917	-0.0000413	-0.0000687	-0.0000132	-0.0001919	-0.0000436
25	-0.0001365	-0.0000673	0.0001192	0.0000006	-0.0000570	-0.0001085
26	-0.0000845	-0.0001824	0.0000503	-0.0000826	0.0000386	-0.0002689
34	0.0000687	0.0000450	-0.0002259	-0.0001742	-0.0000696	0.0000370
35	-0.0001563	-0.0000173	-0.0000413	-0.0001413	0.0001365	0.0000805
36	0.0000561	0.0000134	-0.0000310	0.0000647	-0.0000031	0.0001886
45	-0.0000281	-0.0000507	-0.0000509	-0.0000255	-0.0000094	0.0000087
46	-0.0000461	0.0000845	-0.0001036	0.0000034	0.0000316	0.0002242
56	0.0000309	0.0000129	-0.0000794	-0.0000159	-0.0000653	-0.0000768
111	0.0003602	0.0005507	0.0000119	-0.0001865	-0.0001114	0.0000208
222	0.0002234	-0.0002439	0.0002315	0.0001301	0.0000432	0.0001851
333	0.0002406	-0.0000228	-0.0004838	-0.0004351	-0.0000592	-0.0002898
444	-0.0003904	-0.0001743	-0.0001214	-0.0000760	-0.0000408	-0.0000446
555	-0.0000355	0.0003220	0.0003242	0.0000815	0.0001579	-0.0001373
666	-0.0002795	-0.0001079	-0.0000488	-0.0005163	-0.0002849	0.0003286

Standard errors of estimates:-

R1	R2	R3	R4	R5	R6
0.00027	0.00031	0.00029	0.00028	0.00025	0.00029

Coefficients of multiple correlation:-

R1	R2	R3	R4	R5	R6
1.00000	1.00000	1.00000	1.00000	1.00000	1.00000

Coefficients of partial correlation:-

	R1	R2	R3	R4	R5	R6
H1	1.00000	0.10005	-0.07114	-0.15012	0.09371	0.12863
H2	0.07959	1.00000	0.14260	-0.09251	-0.06523	-0.05072
H3	0.01407	-0.03030	1.00000	0.03405	-0.11368	-0.15832
H4	-0.15845	0.03145	-0.10581	1.00000	0.00207	0.03159
H5	0.00045	0.02708	-0.03647	0.23111	1.00000	-0.04421
H6	0.09803	0.01834	-0.01086	0.13131	-0.15263	1.00000

95% confidence level of the linear coeff.:-

C1-	C2-	C3-	C4-	C5-	C6-
0.0000983	0.0001107	0.0001033	0.0001010	0.0000885	0.0001040
0.0000956	0.0001076	0.0001005	0.0000983	0.0000860	0.0001012
0.0000993	0.0001119	0.0001045	0.0001022	0.0000894	0.0001052
0.0000905	0.0001020	0.0000952	0.0000931	0.0000815	0.0000959
0.0000976	0.0001099	0.0001026	0.0001003	0.0000878	0.0001033
0.0000942	0.0001061	0.0000990	0.0000968	0.0000848	0.0000997

References

[1] Ramaswamy, M. A., Srinivas, T. and Holla, V. S. (1987). A simple method for wind tunnel balance calibration including non-linear interaction terms. *Proceedings of the ICIASF '87 RECORD*, pp. 263–267.

[2] Spiegel, M. R. (1961). *Theory and Problems of Statistics*. Schaum's Outline Series, Schaum Publishing Co., New York.

[3] Yamane, T. (1973). *Statistics, An Introductory Analysis*. 3rd edn, Harper & Row, Publishers, Inc., N.Y., pp. 1080.

DISTRIBUTION

AUSTRALIA

Department of Defence

Defence Central
- Chief Defence Scientist
- FAS Science Corporate Management (shared copy)
- FAS Science Policy (shared copy)
- Director, Departmental Publications
- Counsellor, Defence Science, London (Doc Data Sheet Only)
- Counsellor, Defence Science, Washington (Doc Data Sheet Only)
- S.A. to Thailand MRD (Doc Data Sheet Only)
- S.A. to the DRC (Kuala Lumpur) (Doc Data Sheet Only)
- OIC TRS, Defence Central Library
- Document Exchange Centre, DISB (18 copies)
- Joint Intelligence Organisation
- Librarian H Block, Victoria Barracks, Melbourne
- Director General - Army Development (NSO) (4 copies)
- Defence Industry and Materiel Policy

Aeronautical Research Laboratory
- Director
- Library
- Chief - Flight Mechanics and Propulsion Division
- Head - Flight Mechanics Branch
- Branch File - Flight Mechanics
- Author : S.S.W. Lam

Materials Research Laboratory
- Director/Library

Defence Science & Technology Organisation - Salisbury
- Library

WSRL
- Maritime Systems Division (Sydney)

Navy Office
- Navy Scientific Adviser (3 copies Doc Data sheet only)

Army Office
- Scientific Adviser - Army (Doc Data sheet only)
- Engineering Development Establishment, Library

Air Force Office
- Air Force Scientific Adviser (Doc Data sheet only)
- Aircraft Research and Development Unit
  - Scientific Flight Group
  - Library
- Technical Division Library

Department of Administrative Services
- Bureau of Meteorology, Library

Statutory and State Authorities and Industry

Aero-Space Technologies Australia, Manager/Librarian (2 copies)
Hawker de Havilland Aust Pty Ltd, Victoria, Library
Hawker de Havilland Aust Pty Ltd, Bankstown, Library

Universities and Colleges

Adelaide
Barr Smith Library

Flinders
Library

LaTrobe
Library

Melbourne
Engineering Library

Monash
Hargrave Library

Newcastle
Library

New England
Library

Sydney
Engineering Library

NSW
Physical Sciences Library
Professor R.A. Bryant, Mechanical Engineering
Library, Australian Defence Force Academy

Queensland
Library

Tasmania
Engineering Library

Western Australia
Library
Professor B.J. Stone, Head Mechanical Engineering

RMIT
Library

University College of the Northern Territory
Library

SPARES (10 copies)
TOTAL (73 copies)

AL 149 DEPARTMENT OF DEFENCE

DOCUMENT CONTROL DATA

PAGE CLASSIFICATION: UNCLASSIFIED

PRIVACY MARKING:

1a. AR NUMBER	1b. ESTABLISHMENT NUMBER	2. DOCUMENT DATE	3. TASK NUMBER
AR-005-598	ARL-FLIGHT-MECH-TM-410	FEBRUARY 1989	DST 86/057

4. TITLE

A FORTRAN PROGRAM FOR THE CALCULATION OF THE CALIBRATION COEFFICIENTS OF A SIX-COMPONENT STRAIN GAUGE BALANCE

5. SECURITY CLASSIFICATION (PLACE APPROPRIATE CLASSIFICATION IN BOX(S) IE. SECRET (S), CONF.(C) RESTRICTED (R), UNCLASSIFIED (U)).

DOCUMENT	TITLE	ABSTRACT
U	U	U

6. NO. PAGES: 49

7. NO. REFS.: 3

8. AUTHOR(S)

S.S.W. Lam

9. DOWNGRADING/DELIMITING INSTRUCTIONS

Not applicable.

10. CORPORATE AUTHOR AND ADDRESS

AERONAUTICAL RESEARCH LABORATORY
P.O. BOX 4331, MELBOURNE VIC 3001

11. OFFICE/POSITION RESPONSIBLE FOR:

SPONSOR DSTO
SECURITY -
DOWNGRADING -
APPROVAL CFPD

12. SECONDARY DISTRIBUTION (OF THIS DOCUMENT)

Approved for public release.

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH ASDIS, DEFENCE INFORMATION SERVICES BRANCH, DEPARTMENT OF DEFENCE, CAMPBELL PARK, CANBERRA, ACT 2601

13a. THIS DOCUMENT MAY BE ANNOUNCED IN CATALOGUES AND AWARENESS SERVICES AVAILABLE TO....

No limitations.

13b. CITATION FOR OTHER PURPOSES (IE. CASUAL ANNOUNCEMENT) MAY BE [ ] UNRESTRICTED OR [X] AS FOR 13a.

14. DESCRIPTORS

Strain gauges
Coefficients
FORTRAN 77 (programming language)
Calibrating
Least squares method

15. DRDA SUBJECT CATEGORIES

0094K
0051F

16. ABSTRACT A computer program written in FORTRAN has been developed to calculate the calibration coefficients of a six-component strain gauge balance. The method of Least Squares is used in the calculation procedures which do not require the components of the balance to be loaded independently. The procedures are, in fact, independent of the way the balance is loaded as long as the supplied load/output data provide sufficient information for the determination of the coefficients. The program incorporates several statistical estimations for the evaluation of the performance of the

PAGE CLASSIFICATION
UNCLASSIFIED
PRIVACY MARKING

THIS PAGE IS TO BE USED TO RECORD INFORMATION WHICH IS REQUIRED BY THE ESTABLISHMENT FOR ITS OWN USE BUT WHICH WILL NOT BE ADDED TO THE DISTIS DATA UNLESS SPECIFICALLY REQUESTED.

16. ABSTRACT (CONT.)

calibration process. It can therefore be used as an experimental tool for the investigation of different loading schemes which may be used to calibrate a strain gauge balance.

17. IMPRINT

AERONAUTICAL RESEARCH LABORATORY, MELBOURNE

18. DOCUMENT SERIES AND NUMBER	19. COST CODE	20. TYPE OF REPORT AND PERIOD COVERED
Flight Mechanics Technical Memorandum 410	546411	

21. COMPUTER PROGRAMS USED

22. ESTABLISHMENT FILE REF.(S)

23. ADDITIONAL INFORMATION (AS REQUIRED)